An Overview of the Usage of Default Passwords (extended version)

Brandon Knieriem, Xiaolu Zhang, Philip Levine, Frank Breitinger, and Ibrahim Baggili

Cyber Forensics Research and Education Group (UNHcFREG)
Tagliatela College of Engineering University of New Haven, West Haven CT, 06516, United States
{bknie1, plevi1@unh.newhaven.edu},{XZhang, FBreitinger, IBaggili@newhaven.edu}

Summary. The recent Mirai botnet attack demonstrated the danger of using default passwords and showed it is still a major problem in 2017. In this study we investigated several common applications and their password policies. Specifically, we analyzed if these applications: (1) have default passwords or (2) allow the user to set a weak password (i.e., they do not properly enforce a password policy). In order to understand the developer decision to implement default passwords, we raised this question on many online platforms or contacted professionals. Default passwords are still a significant problem. 61% of applications inspected initially used a default or blank password. When changing the password, 58% allowed a blank password, 35% allowed a weak password of 1 character.

Key words: Default passwords, applications, usage, security

1 Introduction

Security is often disregarded or perceived as optional to the average consumer which can be a drawback. For instance, in October 2016 a large section of the Internet came under attack. This attack was perpetuated by approximately 100,000 Internet of Things (IoT) appliances, refrigerators, and microwaves which were compromised and formed the Mirai botnet. Targets of this attack included *Twitter*, *reddit* and *The New York Times* all of which shut down for hours. The Mirai botnet was created by abusing default credentials in IoT devices [16, 33].

Besides devices, there are also many applications permitting users access to critical central resources such as Database Management Systems (DBMS), Web Server Applications, and Content Management Systems (CMS) which are commonly secured by a username and password. For instance, in July 2014 hackers attacked HealthCare.gov [37]. Fifteen days later HealthCare.gov released a statement that only the test servers were hacked and no personal information was compromised. The attack occurred because the manufacturer's default password on the server had not been changed. This attack was successful because server administrators had neglected this basic security risk. Days later, despite reporting on this vulnerability, the default password had still not been updated [5].

It became imperative to investigate further security vulnerabilities from default passwords. Although in the prior example only a test environment was compromised, commonly test environments go live at some point and during that transition default passwords are often left unchanged.

These findings motivated us to perform two short surveys with the goal to start a discussion in the field about the usage of develop passwords: The first was to examine applications such as DBMS, Web Server Applications, and CMSs that enable a default password during initial configuration. Results show the most applications have default credentials. The second survey was conducted on developers to understand the use of default passwords. The results indicate that many services are designed with default passwords to bypass authentication to provide immediate, temporary access for quick, convenient initial set up of infrastructure and should only be used during this installation phase.

Remark: A short version of this article was published at International Conference on Digital Forensics & Cyber Crime 2017 which published the proceedings in Springer LNICST. If you use this article for your research, please cite the published version.

2 Literature review

"Passwords are an ubiquitous and critical component of many security systems" [38]. Despite the increasing use of biometrics, passwords remain the most common authentication mechanism. Therefore, it is important to create secure passwords that are difficult to compromise. Furthermore, a common theme within information security is that users are the biggest security threat; users can often be careless and fail to prevent fundamental security concerns. As a consequence, administrators often force users to follow password policies. For instance, according to [20], a strong password policy requires a minimum number of characters, different types of characters, and specify how frequently users should change their passwords.

However, [2] found that users may "compromise computer security mechanisms, such as password authentication, both knowing and unknowingly, [... but] such behavior is often caused by the way in which security mechanisms are implemented, and users' lack of knowledge". A result of their survey was that complex policies often force users to write down their passwords, as they cannot memorize them, which creates an additional security concern. This coincides with the findings from [31].

As a consequence, researchers developed additional extensions to enhance password policies. One recommendation is to involve an interface Human-Computer Interaction that observes how humans interact with devices and machines. For instance, [17] concludes that "rather than focusing on password policies on maximizing password strength and enforcing frequency alone, policies should be designed using HCI principles to help the user to set an appropriately strong password in a specific context of use". Another idea is to utilize user's "typing patterns (e.g., duration of keystrokes, and latencies between keystrokes) are combined with the user's password to generate a hardened password" [22]. Practical examples of this include measuring whether or not the user is not only using the appropriate password, but entering it as the expected user should be. I.e., if the user typically enters their password at 140 words per minute, but the entry this time was only 100 words per minute, the user may not be who we expect them to be.

More recently, changes to the National Institutes of Standards and Technology (NIST) Digital Authentication Guidelines have suggestions for improving security and reduce common issues [15]. These suggestions can be broken down into several main categories: hashing passwords, increasing user friendliness, enforcing an 8-character minimum, and banning common passwords. It suggests avoiding password rules, password hints, security questions, and never forcing arbitrary password changes [43]. These suggestions are a step in the right direction for new password policy, though despite being nearly a year old, have yet to be implemented by many application developers.

In the following, the focus will turn exclusively to default passwords. There is additional literature about the aforementioned topics but they are not the main focus of this article.

2.1 Reasons for Default Passwords

Despite widespread use there was little literature on the reasoning for use of default passwords. One argument we found that advocates the use of default passwords was in the field of automation, in a testimony from [39] who states that it is often motivated by vendors with easy remote access and "plant staff is reluctant to change default passwords because of personnel performance considerations during emergency events". Another reason was mentioned by [13] who points out that a benefit of default configurations is ease of usage – "this does make it easier to install new products". However, both references red-flag the usage of default passwords. Following the installation, many are not changed. This is supported by [12], in that users enjoy the convenience of default credentials and are often ignorant on matters of information security.

A justification for using default passwords was asserted by Verifone, a company for electronic payment transactions: "The purpose of this default password is to simply initiate terminal installation, and it is not intended to serve as a strong security control. The important fact to point out is that even knowing this password, sensitive payment information or PII cannot be captured. To date, Verifone has not witnessed any attacks on the security of its terminals based on default passwords. What the password allows someone to do is to configure some settings on the terminal; all executables have to be file signed, and it is not possible to enter malware just by knowing passwords" [10]. However, this statement was released after researchers found the default password of a credit card terminal and published a report about their findings. The report claims Verifone used the same default password for two decades [42].

In summary, there is no substantial, convincing argument that justifies the usage of default passwords. Convenience is often cited but does not provide any form of security. Therefore, it is does not justify use. In fact, the phrase "default password" may as well be an oxymoron; the latter insinuates that it is, at least in part, secure, and this is not the case. Most references attempted to find a good reason for using a default password but subsequently referred to them as a vulnerability and poor practice. Vendors that do not see security concerns as a problem with their product are jeopardizing the safety of their customers. They should know, more than most, that the average consumer likely needs additional help, and they should provide this additional service.

2.2 Breaches Exploiting Default User Credentials

Although guidelines and warnings regarding default passwords exist, there are still many incidents involving default credentials. According to [7], "very few open source vendor advisories have mentioned default passwords, whereas they appear with some regularity in closed source advisories, even in the top 10 [vulnerabilities] as recently as 2005".

A newer study named the Verizon Data Breach Report examined 621 corporate breaches. "The analysis found that 78% of initial intrusions into corporate networks were elementary. Many attackers use a phishing attack, convincing employees to give up credentials, or brute force attack, taking advantage of weak or default passwords on remote services to gain initial access to the network" [40]. Unfortunately, the report did not mention, out of 78% how many constituted weak passwords or default passwords. Notwithstanding, some of the recent breaches that were attributed to the misuse of default passwords:

Utah Department of Health suffered a breach of 780k Medicaid patient health records [27] in addition to compromising more than 255,000 social security numbers [36]. Attackers achieved complete access to the system using a default password.

A Bank of Montreal's ATM was hacked by two 14 year old children; they used the machine's default password [25]. They were able to:

1. Find out how much money was available in the ATM.
2. Change the surcharge amount to one cent.
3. Change the ATM's greeting to "Go Away, this ATM has been hacked".

U.S. Emergency Alert System (EAS) equipment used to broadcast warnings was hacked by exploiting default passwords. After the breach, the hackers sent out an alert warning the public of a 'zombie attack' [25].

Electronic Highway Billboards were attacked in June 2014. The hacker changed the signs to display their Twitter/hacker handle for all the highway drivers to see. This was an act of mischief reported by [18]. According to [23], "the vulnerability is a hard-coded password that could allow unauthorized access to the highway sign, DHS officials said in an alert on Wednesday. Hard-coded passwords, sometimes called back doors, are default logins that software developers code into their programs. The vulnerability was identified in Daktronics Vanguard highway notification sign configuration software".

A recent WordPress incident demonstrated that the usage of a default username can result in a tremendous security risk. In case of WordPress, the default username is always ‘admin’. Hackers used that knowledge and used a botnet to brute force 90,000 IP addresses hosting different software [4]. Unfortunately, the report did not release how successful this attack was.

2.3 Taking advantage of default passwords - tools, scripts and malware

Attackers, often taking an opportunistic approach, realized the potential in abusing default passwords to access a system. Thus, there are several tools, scripts, and malware that can be used for this purpose.

The default password of a given software or product may be easily found online. Furthermore, one can find complete lists of common default passwords; many of which are included in password tool distributions. [35] states “the most common password lists found using Internet search engines were default password lists. These lists contain passwords used by hardware manufacturers as the default security setting”. These lists were, and still are employed by programmers and hackers that use default password scanning tools and worms. [6] mentioned several tools that focus on exploiting default passwords. For instance, Cisco OCS Perl Script scans Cisco devices on a network by inputting ‘cisco’ into the password form. ‘cisco’ is the default password for Cisco routers. Metasploit, a popular penetration testing software tool, includes multiple modules used for network default password scanning. “For instance, the Ektron CMS400.NET Password Scanner module searches for Ektron CMS installations within a network that are using default passwords set up by the vendor”.

On the other hand, several worms exist that use default passwords to propagate. According to [32], the ‘Voyager Alpha Force’ worm was used to demonstrate a vulnerability on Microsoft’s SQL Server with an administrator blank password using the default port: 1433. Similarly, MySQL required no password at the time of installation. A worm named “MySpooler” infected 8000 hosts at a rate of 100 hosts per hour [24]. In 2005, an anonymous developer disclosed a proof-of-concept worm that targeted Oracle databases using default usernames and passwords [44]. A particularly malicious worm implementation uses blending viruses; which are viruses that run a daily Internet scan for vulnerabilities. One of the main functions of them are to find well known default passwords [14].

[34] triggered malnets; a combination of malware and bots initiated malware attacks on routers. Their results demonstrated the little effort required to infect wireless routers using authentication since 25-50% of the users failed to change their default passwords. A similar experiment was also performed by [45]. Regarding wireless malware propagation: 16.7% of routers were set to be configured with default settings. Only 10% of these routers used default passwords or did not have passwords set. The work succeeded in modifying the router firmware.

Even health care applications are vulnerable because of default passwords. They too deal with a plethora of security challenges. [30] suggested conducting regular password audits to evaluate the effectiveness of the health care application security.

The issue of default password management is a repeat offender in system security. There has been a variety of experiments, assumptions, and speculations so as to why users are so reluctant to change these passwords.

2.4 Other devices

Embedded Devices comprise of routers, scanners, printers or IoT devices. An experiment in which a default credential scanner was created using default root credentials for a specific device model [9]. The results of the scanner are as follows:

1. Over 540,000 publicly accessible embedded devices are configured with factory default root passwords. This constitutes over 13% of all discovered embedded devices.
2. 96.75% of accessible vulnerable devices turned out to be still vulnerable.
3. Later the researchers decided to re-test using the same procedure. They found that only 3.25% default credentials had been removed or changed.

It was not programmed to engage in any form of brute force password guessing attacks. Similarly, another study illustrated that out of one million embedded devices fingerprinted 34.2% ran Microsoft IIS and 33.6% ran Apache. In this pool of devices: 2% of the embedded devices, running both IIS and Apache, used default passwords [26].

Higher numbers were discovered by Tripwire; a major cyber threat detection company. They consulted 653 IT or security professionals. Additionally, they consulted 1009 remote workers. All individuals were from either the United States or United Kingdom. Their reports indicate that 30% of IT professionals and 46% of workers fail to change the default password on their wireless routers. Furthermore, 55% and 85% fail to change the default IP address on their routers. This creates opportunities for cross-site request forgery attacks [28]. Expectantly, average users often fail to acknowledge this issue. However, even IT professionals and administrators fail to change these passwords after installation; they become victims as well.

2.5 Databases

Databases have default accounts embedded in them which are prone to vulnerabilities. [3] argues that there are few databases that feature account lockout mechanisms and their authentication process is seldom monitored. Big companies products such as Microsoft SQL and Oracle 9iA are also vulnerable to be attacked by web servers through unaltered default settings [1]. Consequently, the question arises: How many of these database management systems (DBMS) use default configurations? There is a lack of research that seeks to quantify this issue.

3 Problem Description

The widespread use of default usernames and passwords represents a significant threat to both companies and consumers. While security breaches represent a threat to company integrity and their products, they also represent a threat to consumers whose data is stolen. Many default passwords are readily available online and can potentially be used for security breaches. To analyze the impact of default passwords we examined database management systems, Web server applications, and content management systems. This lead to some important questions:

1. How many of these application types utilized default or weak credentials?
2. Next, why do developers permit the use of default credentials?
3. In sum, how can we mitigate security threats from credential negligence?

4 Applications analysis

In choosing applications to test, we decided to focus on web applications as they are easily accessible and cater to a broad audience. More precisely, we investigated Relational Database Management Systems (RDBMS), Web Server Applications (WSA), and Content Management Systems (CMS). Applications were chosen from a list based on popularity, accessibility, cost, and market share [11]. Applications were taken primarily from the most popular, though several average use applications were also tested. Some formerly popular and now deprecated applications were also tested. Of particular note are: Microsoft Access, Oracle RDBMS, PostgreSQL, MySQL, and SQLite as some of the most common applications for DBMS. Microsoft Office is used by 1.2 billion people worldwide [21], and any security failures in their product could affect enormous numbers of users. Microsoft, IBM, and Oracle collectively hold approximately 89% of the DBMS market of 750,000 customers, with Oracle holding around 41.6% (310,000 customers) alone [29].The sheer number of users and the potential compromising information contained in RDBMS, CMS, and WSA software led to the necessity of testing these top applications. Using the aforementioned list containing the major applications in all three categories, our methodology for testing is as follows:

1. For each identified application, search for documentation and identify the default credentials / settings.
2. Download and install a free or evaluation version of each application. Prioritize installation on Windows 10 (64-bit), then Ubuntu Linux 16.04.2, and finally Mac OS Sierra 10.12.5. Use default configurations and procedure; do not use advanced or customized installation options.
3. If a default database is not created during installation, create one immediately after installation.
4. Note any prompts, or lack thereof, regarding security policy enforcement.

5. Assign each conclusive application a password policy quality value on a scale of 0 to 4. This was loosely based on an IBM's classification [41]:

We used Oracle VM VirtualBox 5.1 to create disposable virtual machine environments for testing.

4.1 Results

In total, $n = 90$ applications were analyzed where 62 applications yielded conclusive results and 28[1] had inconclusive results due to licensing restrictions. An overview of the results is given in Tables 1 and 2. Of the 62 conclusive applications, 41 applications had commercial licenses and 21 were open source. To analyze the applications, 51 applications were installed on Windows 10 (64-bit), 8 were installed on Linux-x86, four were web services, and one was installed on Mac OS. Note, two applications were a pre-release version (0.1 - 0.9/Alpha/-Beta), the remaining 60 applications were a release version (1.0+) (97%).

Default passwords. In total, 30 applications featured a default user name, the most frequent were “Admin” or “root”. 6 (10%) applications featured a default password. 32 (52%) applications featured a default blank password for the default user account. All applications featuring a default password also featured a default user name.

Password policy quality. Lastly, we analyzed the quality of the passwords according the IBM classification [41] which is divided into:

Level 0 - No password policy.
Level 1 - Weakest password policy; only requires a single character.
Level 2 - Requires a minimum number of characters but can be compromised without the aid of a computer.
Level 3 - Requires a minimum number of characters but can still likely be compromised with the aid of a computer.
Level 4 - Requires a minimum number of characters, numbers, and special characters, and would be difficult, but not impossible, to compromise; even with the aid of a computer.

Overall, 36 (58%) applications were categorized as having a level 0 policy, 22 (35%) applications were categorized as having a level 1 policy. Two applications were categorized as having a level 2 policy. One application was categorized as having a level 3 policy. Finally, only one application that met the requirements for a level 4 policy, which is interesting as this is what most modern online portals require.

[1] Actian Ingres, Actian Vector, CA Datacom, CA IDMS, Clarion, Clustrix, Empress Embedded Database, EXASolution, eXtremeDB, GroveSite, IBM PureSystems, Infobright, Linter, Microsoft Visual FoxPro, NexusDB V4 Windows, NonStop SQL, Openbase, Postgres Plus Advanced Server, R:Base, SAP ADS, SAP Anywhere, SAP HANA, SAP Sybase ASE, SAP Sybase IQ, SQL Azure, SQream DB, UniData, Vertica

Summary. There are applications found in our study that feature both default account names and passwords for the application administrator account. Most of the applications administrator's account user names had default values of "root" and "admin". Alternatively, applications often used the application name in the default credentials. Several RDBMS applications use identical values for both the user name and password.

5 Qualitative survey of default credential use

This section tries to understand why default user credentials / passwords are still so widely used. Therefore, we acquired IRB[2] approval and asked software developers, computer engineers, and security experts for insight:

> Why do many applications still come with a default user name password and do not require the user to set new credentials according to a reliable password policy?

The question was distributed online in 20 software developer forums, advertised to 30 groups on Quora, and other forums. The question was also sent directly to 35 users on Quora who are known developers. Besides the online distribution, we sent it to 10 professors from the University of New Haven and the University of Bridgeport.

Summary of findings. The question received high exposure; in one instance over 2,800 individuals accessed or viewed the question on Quora. However, the response rate was low. In total, we only received 20 responses. The answers can be summarized as follows:

- 6 users blamed the developers for writing a sloppy code.
- A Web Development project manager on Quora described a situation: "I ran across a custom WordPress / Yii app that used the same password by default. As the dev manager, I pointed out that this was a major flaw. Got told that it was but wasn't urgent. Until a hack happened. By then I'd redesigned the password system with all sorts of goodies that included rate limits on attempts, no password access, even for admins, and all sorts of other stuff".
- The CEO of mid-size online company on LinkedIn explained a situation where a default password is used: "I need to install my Lazarus application on 20 clients. Can you imaging running through the setup process with password policies right from the start? Do you see how much more time you'll need to spend? ... I imagine you know the hassle of dealing with OS permissions, DB permissions (different user), application permissions, and then user roles. Yes, it is possible to have a security policy in place from the start, but do you see how much more difficult it gets?"

[2] We obtained a category two exemption from the Institutional Review Board (IRB) at the University of New Haven restricting the survey from recording participant identification information or behavior, and disclaiming that it posed risk or harm to subjects not encountered in everyday life.

The sample size was not substantial and therefore we cannot claim it is representative of the whole. However, it did yield some direction, however minor, that will be considered for discussion or used in resulting studies.

6 Discussion

Default user credentials. While many web portals already enforce strong password policies, some applications are still lacking; default user credentials are still prevalent. While several individuals blamed developers for writing ”sloppy” code, we believe these are design decisions were purposefully made in favor of other more pressing features. Alternatively, developers may not want to affect the status quo. Default credentials have been a staple configuration policy for decades, so why change it? Lastly, developers may simply not be aware of the importance of security policies; a dangerous mix of when it comes to widespread product use. Our findings are not new. Documentation for default passwords is readily available online for a variety of applications; many of which beyond the scope of this article. Attackers use that knowledge and feed default user credential password dictionaries to their password cracking tools. For instance, Hydra or Jack the Ripper usually come with default dictionaries which contain frequently used credentials

On the other hand, applications are designed to provide the best user experience to their customers and reduce setup time. Streamlining the installation process contributes to this effort. It is faster than typing even a documented default password or generating one for each instance of the application; especially when the administrator needs to install the application on multiple devices in succession. The default passwords in this study demonstrate this by being easy to remember and utilize for multiple devices. For instance, most of applications used ‘password’, ‘admin’, ‘dba’ etc as default passwords. In addition, after installation it is far easier for a system administrator to access software using a default password as they need not remember the specific credentials assigned to the application.

Weak Passwords. Many of these applications accepted a single character as a valid user name or password. A user may choose a more complex password, but because there is often no requirement for special characters or total character count, the user may choose the easiest, most convenient credential solution. While easy, this solution is often the least secure. That attackers use that knowledge is frequently demonstrated in the media. For instance, a recent breach suffered by Ashley Madison in July 2015 compromised customer information. The breach affected 30 million users and exposed 60 GB of data. The website failed to integrate a substantial password policy. A group named Cynosure Prime posted passwords for 11.7 million users after the breach and the most common passwords used were “123456”,“12345” and “DEFAULT” [8]. A study by the Hamilton Institute concurred with this and found additional common passwords, “password”, “iloveyou”, “princess”, and “rockyou” [19]. These simple, low-entropy passwords represent a significant threat to their users.

Users. The user remains a significant threat to their own security. Applications sometimes include a password strength indicator. This shows the user how weak or strong their chosen password input is before committing to it. However, it is still up to the user, not a dedicated password policy, to ultimately select a password of appropriate strength, as low strength passwords are often accepted.

Average users may not prioritize cyber security concerns. These users may not feel it is necessary to secure their software because they do not use it to store sensitive information, are not aware of necessary diligence, or do not care. All of the examined applications provide a means to change user credentials. Therefore, the user should appreciate the need to have a secure password that is changed often or begin investing in alternative, non-string password solutions. In this case the application may not be to blame the user for a compromising event because it assists with and provides advisory statements with regards to post-installation credential management that were not properly utilized by the user. However, the user can also blame the application for not enforcing a substantial password policy. Displacement of security due diligence is an increasing concern. Yes, the consumer should have the freedom to choose, but at what point is the developer doing the consumer a disservice?

At what point does the user take responsibility for their security? To what degree is the application developer responsible for forcing users to be more secure? As of now it appears that neither the producer nor consumer have considered due diligence regarding the matter. The default password dilemma devolves into a self-fulfilling prophecy. If developers continue to forgo proper password policies then that is what will continue to be expected by users. Users will then not utilize more secure measures, as they will have become accustomed to poor security credential management. This in turn will prevent application developers increasing security measures.

6.1 Possible Solutions

Password policies, as we know them today, are not substantial enough to withstand targeted attacks. Instead, we propose a series of solutions that would greatly reduce risk at very little cost to the administrator.

Ideally, universally enforcing password policies, no matter the medium, would encourage users to be more security minded from the start. Additionally, education/training may assist in reforming this poisonous culture. Training may come from within in order to create a better product or be government incentivized via standard requirements. Especially within software applications, password policies need to become the rule; not the exception. This could be achieved with either legal or commercial standardization practices. We offer some solutions and alternatives to mitigate the potential for default and weak password abuse:

– Applications should require the user to set up customized credentials upon installation. Better yet the application itself should not be allowed to function

properly without these customized credentials. During the choice of the password, a password policy should be enforced, or at least a password strength indicator should be shown. This standard can be enforced by not allowing the user to proceed until they meet a certain string password strength requirement.
- Applications could become smart and warn the administrator when it goes live that default user credentials are still in place. This policy could be perpetuated using different, use appropriate licensing. Alternatively, the software may have two separate functioning modes; one production, one live. Toggling from production to live may prompt this check.
- Advances in security technology have lead to widespread use of dual layer authentication. Dual layer, also called two-factor or two-step authentication, requires the user to have two separate means of authentication in order to access a service. Gmail, Apple, Facebook, Twitter, PayPal, Dropbox, Steam, and many more applications already employ this security policy. If dual layer authentication is not included in the current product then it should be as most people have an email address or mobile number.

We acknowledge that mandatory policy standards would be more difficult to integrate as it would require widespread acceptance and support. However, there is incentive for both for-profit and non-profit parties:

For-profit solutions involve creating marketing opportunities to outperform the competition in the security sector. However, we also suggest that developers use an enhanced credential feature set to their advantage by marketing it as prestigious. For example, a seal of approval on each product identifying the lengths at which the developers went to assure security due diligence. Intel and Windows both often label laptops from various brands to promote their product. Energy saving labels are prominently displayed to demonstrate a product's cost savings. Why not utilize a "Security Certified" label as well? This way, we appeal to business sensibilities; many developers are inevitably interested in financial growth. Businesses may begin to clamber to market their own products as more secure than their competitors; we need to make security sexy.

Non-profit, or government funded solutions, create opportunities for control over security standards. Worst case, commercial entities that do not meet these standards may lack the endorsement to sell product until they meet certain requirements; in much the same way as Volkswagen needed to reevaluate their emission standards or the Environmental Protection Agency requires businesses to dispose of hazardous waste in an appropriate manner.

Going forward, developers should consider foregoing string passwords as we know them today in favor of biometric solutions. Fingerprint scanners are already featured on many mobile devices. Retinal scanners are also a less mainstream possibility that may trump insecure facial recognition techniques.

Marrying both ideas, dual layer authentication could also incorporate keystroke dynamics or an accelerometer. That is, either exclusively using a string password or in conjunction with a biometric password, also measuring the integrity of the user's attempt. For example, the password may be correct, but the user's haptic interaction must also be conclusive according to the stored baseline that was measured during credential configuration.

Lastly, there are opportunities in default passwords in other security applications; to deceive potential attackers by using a 'honeypot' mechanism. The actual application data should be protected by a secure password policy solution. However, purposely leaving a separate, vulnerable application without a responsible password policy may allow administrators to detect and catch potential threats.

6.2 Developers and Default Credentials

We have demonstrated the prevalence of default password use but need to revisit the question "Why are developers still using default passwords?". We have to acknowledge both the user and developer sides of the argument. Our brief survey can be summarized as follows:

- Users blame developers for writing sloppy code. This could be expanded into a discussion about developer blaming concerning a lack of due diligence.
- Developers may not prioritize a credential security feature set over other pressing issues. One developer acknowledged the security flaw but it was not an urgent enough feature set to prioritize. Developers may also feel as though security due diligence should be the responsibility of the user.
- Lastly, adhering to a more substantial credential policy is inconvenient. With regards to the security triangle, confidentiality, integrity, and accessibility, the more secure a policy, the less accessible it becomes. When a system administrator is configuring twenty different installations they may not want to bother with a more substantial policy to save time.

The survey results can be further summarized. Both the users and developers seem interested in displacing the responsibility of security due diligence. Users are also interested in minimizing effort, especially in larger installations or management, in order to make their responsibilities more accessible in the future.

7 Limitations

There are some notable limitations in this study: First, a large portion of the applications found were inaccessible for downloading and installing because they were not open-source, expensive, or came with demo or trial versions that did not contain login information required for this research. Therefore, it was often necessary to use documentation for information on default settings. Additionally, we could not follow the same installation procedure for all applications as all are installed differently. Secondly, we received a statistically insignificant number of survey responses. While the responses we received encourage further discussion, they are not suitable for drawing conclusions. Future research should focus, in part, on finding a more reliable means of surveying professionals. Lastly, all tests were done manually and therefore prone to human errors.

8 Conclusion

This article surveyed a well-known default password issue on 21 open-sourced applications and 41 commercial applications. Out of the 62 applications, we found that 32 applications featured a default user name, 6 applications featured a default password and 32 applications accepted empty passwords. In total, 38 applications surveyed can lead an administrator using default user credentials. Meanwhile, in order to evaluate the password policy we also scored the applications with IBM password quality scale. 36 of applications scored with '0', having no password policy. 22 of applications scored a '1', meaning that a single character password is acceptable, the weakest possible password policy. Only 4 applications had an acceptable password policy. To explain why practitioners may keep default user credentials of the DBMS on their own database system, we distributed a survey on Quora and responded by variety roles such as web developer, system manager, CEO etc (Sec. 5). Precisely, the reasons could be developer's negligence, complexity of the setup, lack of management etc. Besides, we further discussed the reasons in depth and put forward some solutions in accordance with the default-password issues.

Acknowledgements

Special thanks go to Mohammed Nasir who initially started this research project and Matthew Vastarelli for supporting us.

References

1. Database insecurities: Are hackers winning by default? *Info-Tech Advisor Newsletter*, page 1, Jun 24 2002. Copyright - Copyright Information Technology Research Group, Inc. Jun 24, 2002; Last updated - 2014-05-19.
2. Anne Adams and Martina Angela Sasse. Users are not the enemy. *Communications of the ACM*, 42(12):40–46, 1999.
3. Rob Barnes and Enterprise Auditing Solutions Director. Database security and auditing: Leading practices. 2011.
4. Logan Booker. Brute Force Attack Targets WordPress Sites With Default Admin Username, 2013.
5. Rebecca Carroll. Breached healthcare.gov server still had default password, 2014.
6. Brad Casey. Network security risks: The trouble with default passwords, 2014.
7. Steve Christey and Robert A Martin. Vulnerability type distributions in cve. *Mitre report, May*, 2007.
8. Keith Collins. The top 100 passwords on ashley madison, 2015.
9. Ang Cui and Salvatore J Stolfo. A quantitative analysis of the insecurity of embedded network devices: results of a wide-area scan. In *Proceedings of the 26th Annual Computer Security Applications Conference*, pages 97–106. ACM, 2010.
10. DataBreaches.net. Verifone statement on default password z66831, 2015.
11. DB-Engines. Db-engines ranking. 2017.

12. Sokolov Dini. Internet of things security problems. *[RUS] "Current Information Protection"*, (1):1–8, 2017.
13. Barbara Y Fraser. Site security handbook. 1997.
14. John Gordineer. Blended threats: A new era in anti-virus protection. *Information Systems Security*, 12(3):45–47, 2003.
15. Garcia Grassi. Digital identity guidelines. *National Institute of Standards and Technology*, 2016.
16. Nyman Hypponen. The internet of (vulnerable) things: On hypponen's law, security engineering, and iot legislation. *Technology Innovation Management Review*, 7(4):5–11, Apr 2017.
17. Philip G Inglesant and M Angela Sasse. The true cost of unusable password policies: password use in the wild. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 383–392. ACM, 2010.
18. KrebsonSecurity.com. They hack because they can, 2014.
19. Mayer Malone. Investigating the distribution of password choices. *Hamilton Institute*, pages 1–14, 2011.
20. Flavio Martins. Creating strong password policy best practices, 2014.
21. Microsoft. Microsoft by the numbers. 2017.
22. Fabian Monrose, Michael K Reiter, and Susanne Wetzel. Password hardening based on keystroke dynamics. *International Journal of Information Security*, 1(2):69–83, 2002.
23. Nextgov.com. Flaw lets hackers control electronic highway billboards, 2014.
24. Stephen Northcutt. The risk of default passwords, 2007.
25. Thu Pham. Default passwords: Breaching atms, highway signs & pos devices, 2014.
26. Mathew J. Schwartz. Corporate espionage's new friend: Embedded web servers. *Informationweek - Online*, Sep 26 2011. Copyright - Copyright 2011 CMP Media LLC. All rights reserved. No part of the report or the data or information included therein may be reproduced, republished or redistributed without the prior written consent of CMP Media LLC; Last updated - 2011-09-27.
27. Duo Security. Utah department of health (udoh) breach, 2012.
28. Info Security. 80% of soho routers contain vulnerabilities, 2014.
29. Anne Shields. Will oracle position itself as a key cloud player in 2017? 2017.
30. Vince Stanford. Pervasive health care applications face tough security challenges. *pervasive computing, IEEE*, 1(2):8–12, 2002.
31. Wayne C Summers and Edward Bosworth. Password policy: the good, the bad, and the ugly. In *Proceedings of the winter international synposium on Information and communication technologies*, pages 1–6. Trinity College Dublin, 2004.
32. Microsoft Customer Support. An unsecured sql server server that has a blank (null) system administrator password allows vulnerability to a worm, 2005.
33. Symantec Security Response. Mirai: what you need to know about the botnet behind recent major ddos attacks, Oct 2016.
34. Patrick Traynor, Kevin Butler, William Enck, Patrick McDaniel, and Kevin Borders. malnets: large-scale malicious networks via compromised wireless access points. *Security and Communication Networks*, 3(2-3):102–113, 2010.
35. RP Van Heerden and JS Vorster. Statistical analysis of large passwords lists, used to optimize brute force attacks. 2009.
36. JaiKumar Vijayan. Weak passwords still the downfall of enterprise security, 2012.
37. Kate Vinton. Data breach bulletin: Home depot, healthcare.gov, jp morgan, 2014.
38. Kim-Phuong L Vu, Robert W Proctor, Abhilasha Bhargav-Spantzel, Bik-Lam Belin Tai, Joshua Cook, and E Eugene Schultz. Improving password security and

memorability to protect personal and organizational information. *International Journal of Human-Computer Studies*, 65(8):744–757, 2007.

39. Joseph M Weiss. Control systems cybersecurity–maintaining the reliability of the critical infrastructure. *Testimony of Joseph M. Weiss Control Systems Cybersecurity Expert before the House Government Reform Committee?s Subcommittee on Technology, Information Policy, Intergovernmental Relations, and the Census US House of Representatives*, 2004.
40. Robert Westervelt. Verizon data breach report finds employees at core of most attacks, 2013.
41. Christie Williams and Katherine Spanbauer. Understanding password quality, 2001.
42. Martyn Williams. Researchers reveal the default password for a credit card terminal, 2015.
43. Wisniewski. *Naked Security*, 2016.
44. Joshua Wright. Oracle worm proof-of-concept, 2005.
45. Stefano Zanero. Wireless malware propagation: A reality check. *Security & Privacy, IEEE*, 7(5):70–74, 2009.

Table 1. Surveyed Applications

Name	**Version/ Release**	**Platform**	**Commercial/ Open-Source**	**License**	**Default Username**	**Default Password**	**Password Policy Quality**
4th Dimension	16.1	Windows	Commercial	30-Day Evaluation	“Administrator”	None	0
Adabas	2016 April	Windows	Commercial	Community Edition	Inherits user account.	Inherits user password.	0
Alpha Five	V12	Windows	Commercial	30-Day Evaluation	“Admin”	None	0
Altibase	6.5	Linux-x86	Commercial	Community Edition	None	None	0
Amazon Aurora	N/A	Web-Service	Commercial	N/A	None	None	2
Apache Derby	10.13.1.1	Windows	Open-Source	N/A	None	None	0
Apache OpenOffice.org Base	4.1.3	Windows	Open-Source	N/A	N/A	N/A	0
Apache Trafodion	2.1.0	Windows	Open-Source	N/A	None	None	1[2]
Base X	8.6	Windows	Open-Source	Free Version	”admin”	”admin”	0
ClickHouse	1.1.54189	Linux-x86	Open-Source	N/A	None	None	0
CSQL	3.3	Linux-x86	Open-Source	N/A	None	None	0
CUBRID	10.0.0.1376	Windows	Open-Source	N/A	“admin”	“admin”	2[3]
Database Management Library (C++)	1.0	Windows	Open-Source	N/A	None	None	0
DataEase	6.5 Demo	Windows	Commercial	N/A	“labadmin”	None	0
Dataphor	3.1.6143	Windows	Open-Source	N/A	“admin”	None	0
dBase PLUS	11.2	Windows	Commercial	30-Day Evaluation	None	None	0
Drupal	8.3.2	Windows	Commercial	Free Version	None	None	1
EnterpriseDB	9.6	Windows	Commercial	Standard Version	“postgresql”	None	1
FileMaker Pro	15	Windows	Commercial	Trial Version	“Admin”	None	0
Firebird	3.0.2	Windows	Open-Source	N/A	N/A	N/A	1
FrontBase	8.28	Windows	Commercial	Free Version	None	None	0
Google Fusion Tables	N/A	Web Service	Commercial	Free Version	Google Account	Google Account	3[4]
Greenplum	5.0.0-alpha.3	Linux-x86	Open-Source	N/A	None	None	0
H2	1.4.195	Windows	Open-Source	N/A	“sa”	None	0
Helix	7.0.2	Mac OS	Commercial	Demo Version	None	None	0
HSQL	2.4.0	Windows	Open-Source	N/A	“SA”	None	0
IBM DB2	11.1	Windows	Commercial	Trial Version	“db2admin”	None	1
IBM DB2 Express-C	11.1	Windows	Commercial	Trial Version	“db2admin”	None	1
Informix Enterprise	12.10	Windows	Commercial	Time-Limited	“informix”, “ifxjson”	None	0
InterBase	2017	Windows	Commercial	Trial Version	“SYSDBA”	N/A	1

0: No password policy.
1: Password policy only requires a single character.
2: Requires a minimum number of characters but can be compromised without a computer.
3: Requires a minimum number of characters but can still likely be compromised with a computer.
4: Requires a minimum number of characters, numbers, and special characters, and would be difficult to compromise.
[1]: Fully custom credentials required.
[2]: Forces custom credentials following login with defaults.
[3]: Two-factor authentication required.

Table 2. Surveyed Applications (Continued)

Name	Version/ Release	Platform	Commercial/ Open-Source	License	Default Username	Default Password	Password Policy Quality
InterSystems CachÃ©	2017.1	Windows	Commercial	Evaluation Version	“_SYSTEM”, “Admin”, “SuperUser”, “forensics”, “CSPSystem”	N/A	1[2]
JBoss Web Console	6	Windows	Commercial	Free Version	“Admin”	“Admin”	0
Joomla	3.7	Windows	Commercial	Free Version	“admin”	None	1
LibreOffice Base	5.3.3	Windows	Open-Source	N/A	None	None	0
MariaDB	10.3	Windows	Open-Source	Free Version	“root”	N/A	1
Microsoft Access	16.0	Windows	Commercial	Office 2016	None	None	0
Microsoft SQL Server	2016 SP1	Windows	Commercial	Express Edition	“sa”	None	0
Mimer SQL	10.1	Windows	Commercial	Trial Version	“SYSADM”	N/A	1
MonetDB	11.25.21	Windows	Open-Source	Free Version	None	None	0
mSQL		Linux-x86	Commercial	Free Version	“root”	None	0
MySQL	5.7.18.1	Windows	Commercial	Community Edition	“root”	None	0
neo4j	3.2	Windows	Commercial	Evaluation	“neo4j”	None	1[1]
NexusDB	V4	Windows	Commercial	Server Trial Version	N/A	N/A	1
NuoDB Database	2.6.1	Windows	Commercial	Community Edition	“dba”	“goalie”	1[1]
NuoDB Domain		Web Service	Commercial	Community Edition	None	None	1
OpenLink Virtuoso	6.0	Windows	Commerical	Trial Version	N/A	N/A	1
Oracle RDBMS	7.3	Windows	Commerical	Free Version	N/A	N/A	0
Oracle TimesTen		Windows	Commercial	Free Version	N/A	N/A	1[2]
Orange HRM	3.3.1	Windows	Open-Source	N/A	None	None	1[2]
Polyhedra	8.6.1	Windows	Commercial	Lite Version	None	None	0
PostgreSQL	9.6	Windows	Open-Source	N/A	”postgres”	None	0
RDM Server	8.4	Windows	Commercial	Trial Version	N/A	N/A	1[2]
SAND CDBMS	8.1	Windows	Commercial	Free Version	“DBA”	None	0
SAP MaxDB	7.8.02.39	Windows	Commercial	Free	“DBADMIN”	N/A	1
ScimoreDB	4.0	Windows	Commercial	Freeware	None	None	0
SQLBase	12.0	Windows	Commercial	Trial Version	“SERVER1”	“SECRET”	0
SQLite	3.18	Windows	Open-Source	N/A	None	None	0
Tableau (Local)	10.2.2 64-bit	Windows	Commercial	14-Day Evaluation	N/A	N/A	0
Tableau (Online)	10.2.2 64-bit	Windows	Commercial	14-Day Evaluation	N/A	N/A	4
Tibero	6.0	Windows	Commercial	30-Day Evaluation	“root”, “sys”, “syscat”, “sysgis”, “outln”, “tibero”, “tibero1”	“tibero”, “tibero”, “syscat”, “sysgis”, “outln”, “tmax”, “tmax”	1
txtSQL	3.0.0b	Windows	Open-Source	N/A	“root”	None	0
Wordpress	4.7.4	Web Service	Open-Source	N/A	None	None	1

0: No password policy.
1: Password policy only requires a single character.
2: Requires a minimum number of characters but can be compromised without a computer.
3: Requires a minimum number of characters but can still likely be compromised with a computer.
4: Requires a minimum number of characters, numbers, and special characters, and would be difficult to compromise.
[1]: Fully custom credentials required.
[2]: Forces custom credentials following login with defaults.
[3]: Two-factor authentication required.